International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 09 | Sep 2020 www.irjet.net p-ISSN: 2395-0072

Integrating Virtual Voice Assistant and MQTT in IoT based Smart Home Application

Ms. Pooja Pingal[1], Dr. Vivek Kumar[2], Mr. Vikrant Verma[3]

[1]*M.Tech. Scholar, Department of Electronics & Communication Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India*
[2]*Head of Department of Electrical & Electronics Engineering, B.R.C.M.C.E.T., Bahal, Haryana, India*
[3]*Embedded System Design Engineer, Elecsys India Pvt. Ltd., Kalka, Haryana, India*

Abstract - *In this paper an IoT based smart-home system was designed and implemented. The sole purpose was to empower the user to monitor the status and control the switching of multiple electrical appliances remotely either via some voice based commands or through a GUI based application over the internet. Here for the voice based control using smartphone a Voice Personal Assistant (VPA) by Google i.e. Google Assistant was used. For GUI based monitoring and control an MQTT broker by Adafruit i.e. AdafruitIO was used. The status of relays could be observed over the AdafruitIO dashboard on a laptop or mobile phone from anywhere in the world. Another purpose was to integrate and connect these two different services via an 'IFTTT' Applet so that they could be able to communicate with each other. So, here in the IFTTT applet created, 'Google Assistant' was used as a Trigger service whereas 'AdafruitIO' was used as an Action service for this system. The system was built around NodeMCU computational board.*

Key Words: MQTT, IFTTT, IoT, VPA, Google Assistant, AdafruitIO, WiFi, NodeMCU, etc.

1. INTRODUCTION

As the technology is advancing and changing at a rapid pace the trend of human interaction with the machines is evolving day by day. Earlier this process started with the buttons and slowly this trend of human-machine interaction shifted to the touch-pads and now this trend has reached a level where humans are commanding intelligent machines by just talking to them. It means now humans can control machines by giving voice commands in which Artificial Intelligence plays a vital role in the realization of natural dialogue between humans and machines. In recent years, the voice systems, also known as interactive conversational systems are the fastest growing area in AI. These are known as Voice Assistant (VA) or Virtual Personal Assistant (VPA) system and can be used in different areas of applications, including education assistance, medical assistance, robotics and vehicles, disabilities systems, home automation, and security access control. Nowadays, Alexa, Siri, Google Assistant are becoming more and more trendy virtual personal assistants and the developers are frequently trying to exploit and integrate their features in smart devices and applications.

2. MOTIVATION

The motivation behind this effort came from the consistently rising claim for smart homes and people's orientation towards adopting more and more Artificial Intelligence and IoT based systems in their businesses and day to day life. These cutting edge technologies have the power to strongly influence the working methodologies of humans as well as machines. The Human Machine Interface (HMI) can do a lot in some specific areas like robotics, smart homes, smart cars, etc. Further, for developers, the easy access to some of the open source platforms like VPAs including Google Assistant, Alexa, various MQTT platforms like Adafruit IO, Blynk, ThingSpeak, integration of two different apps via IFTTT, etc. has increased the spectrum of innovation worldwide. Even hobbyists can be seen using these platforms for developing their freaking gadgets. Although these platforms are limited in terms of their free usage but provides an opportunity to enhance one's knowledge level by exploring the potentials of these applications.

3. OBJECTIVE OF STUDY

The objective of the study was the implementation of an IoT based multi-appliance control system remotely through phone via interactive voice commands using a VPA (Virtual Personal Assistant) by Google i.e. Google Assistant. In other words, a voice controlled home automation system needs to be designed and implemented. The proposed system should be capable enough to recognize the voice based commands given by the user and respond back through voice as well as text message. The command provided by the user to the VPA at any instant should match the already saved command in the form of some text phrase into the cloud and enable the user to control anything remotely if there is a match. The VPA should respond to the user by converting this voice command-to-text and revert back to the user by replying in voice and text both and also concurrently by performing the assigned task or transferring the commands to the device.

4. AIM OF STUDY

The aim of this work was to design and develop a hardware prototype board deploying 4-channel relay

switch circuit. These relays were to control four different electrical appliances remotely by the user via smartphone having internet connection and an AI based Virtual Personal Assistants (VPAs) by Google i.e. "Google Assistant" installed in it. The IoT system was developed around NodeMCU development board and the connection established between the smartphone device and the prototype board was wireless only using the local Wi-Fi hotspot. Another aim was to enable the user to control these relays globally from any place via an MQTT dashboard i.e. AdafruitIO installed on a laptop or smartphone. The status of the relays on the AdafruitIO dashboard should be in synchronism to the relay switching commands made over the Google Assistant. The hardware prototype and MQTT dashboard status should validate the work.

5. SYSTEM WORKING PRINCIPLE

This project uses two services to make it control through Google Assistant from anywhere in the world, Adafruit MQTT and IFTTT. Adafruit MQTT broker allows changing the message feed/ topic from any internet connected device globally. Here, the NodeMCU was acting as a MQTT client hence it was constantly listening to Adafruit MQTT broker. So if any changes occur in the server side, the same changes will be observed on the client side i.e. on our NodeMCU board. And to change our message on the MQTT broker side via Google assistant, we were using one service called IFTTT. In IFTTT, we were making an applet in which we could connect two services, Google Assistant and Adafruit MQTT. So by making proper applet, we could successfully update the message feed/ topic on the adafruit broker side with Google Assistant on the phone. If This Then That, also known as IFTTT is a freeware web-based service that creates chains of simple conditional statements, called applets. An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, or Pinterest. IFTTT pairs two different services/ devices so that they can talk to each other. In other words, IFTTT is the free way to get all the apps and devices talking to each other.

6. PROBLEM STATEMENT

1. A high-performance, low-cost IoT-integrated computational platform need to be selected. Here, NodeMCU development board deploying ESP8266 Wi-Fi chip was chosen for the implementation of proposed work
2. An investigation was carried out to identify the best out of available MQTT platforms and finally AdafruitIO was considered as the best suited MQTT platform for the implementation of proposed work
3. Gained significant knowledge about the potentials and utilities of NodeMCU, Wi-Fi protocol, MQTT protocol, IFTTT, Google Assistant VPA, relay and its driver circuit and also about their interfacing with the microcontroller
4. Surveyed the literature and other related works submitted by others for exploring the potentials and proper methodologies to implement the task and inferences were drawn out of it
5. Designed working flow diagram and algorithm to implement the proposed work
6. Designed and developed the hardware board around a computational node and written firmware for that using an IDE. Here, the computational node was NodeMCU and the firmware was written in Arduino IDE
7. Performed multiple iterations to test and calibrate the system and finally implemented it

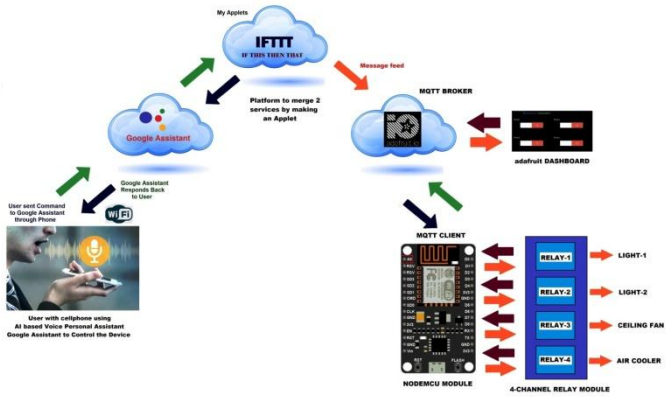

Fig.-1: System Architecture

7. IMPLEMENTED WORK

The implemented work was an IoT (Internet-of-Things) based home automation system to control multiple electrical appliances by accepting voice input commands from a VPA in the smartphone using local Wi-Fi hotspot network. Here the work was implemented over the Google Assistant VPA. Also the system was handled by an MQTT Dashboard i.e. AdafruitIO which enables the user to monitor the status and control the switching of appliances from anywhere in the world via internet. The system was implemented around the NodeMCU development board which provided general purpose input/ output pins for the interfacing of relays and an embedded Wi-Fi node ESP8266 to connect the hardware system to the internet. The D2, D3, D5, D6 digital input output pins of NodeMCU were connected to the 4-channel SPDT relay module. The relays are electromagnetic switches and were connected to the ac load. The load could be a bulb, CFL, fluorescent tubelight, LED bulb, fan, air cooler, motor, etc. Here, for the demonstration purpose we had connected four LED bulbs to represent the ac load. The system was made to operate on a 5V / 2A DC switched mode power supply.

8. METHODOLOGY ADOPTED

Here in this section the methodologies followed to design and implement this system using different hardware and software tools were discussed.

8.1 System Components

To build this application, these different hardware components were used

- NodeMCU –ESP8266 development board with Wi-Fi SoC.
- 4-Channel Relay Board
- +5V DC power supply

Also, to build this application, three different software platforms were used

- 'Google Assistant' Virtual Personal Assistant
- 'AdafruitIO' MQTT Broker
- 'IFTTT' Applet

To use above services we need to configure these software platforms individually.

8.2 Experimental Set-up

The experimental setup was developed as per the circuit design and as shown below a four-channel SPDT relay circuit board was designed and deployed in the system along with a mini buzzer and a regulated DC +5Volt power supply. The output of these relay modules were connected to switch the ac load. For demonstrating the ac-load four LED bulbs were used here. Otherwise, any single phase ac electrical appliance could be connected with relay outputs.

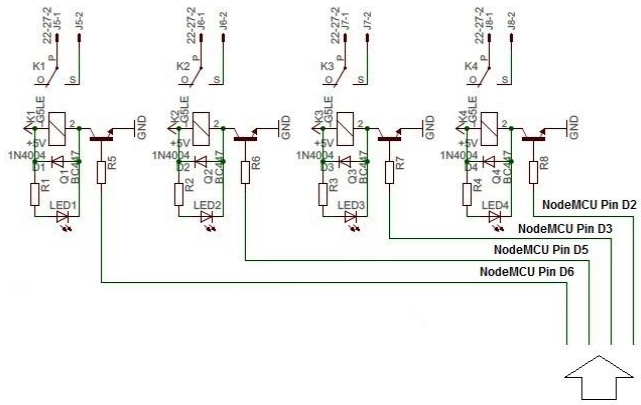

Fig.-2: Four-Channel Relay Circuit

This relay board was interfaced to the NodeMCU development board. NodeMCU was the computational platform used in this system as it was responsible to gather information, analyze it, assign commands, take decisions and take actions for the system. NodeMCU is a low-cost open source IoT platform. It is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). It runs on the ESP8266 Wi-Fi SoC (System on Chip) from Espressif Systems. Both the firmware and prototyping board designs are open source. ESP-12F Wi-Fi module is having a core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates industry-leading ultra low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi on-board antenna. The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol stack. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement. ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any microcontroller based design with simple connectivity (SPI/SDIO or I2C/UART interface). ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs.

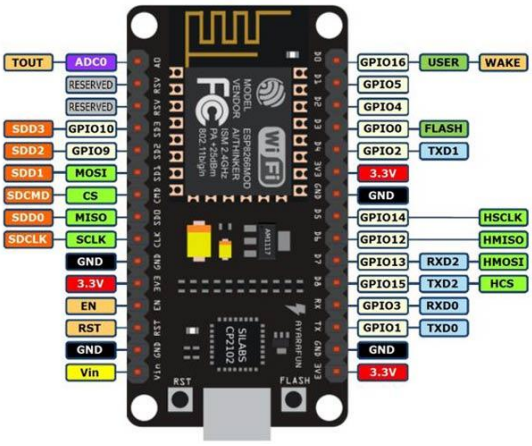

Fig.-3: NodeMCU Pin Configuration

9. FIRMWARE COMPONENTS

The firmware was edited and compiled in the Arduino IDE (Integrated Development Environment). Initially the necessary libraries were downloaded and managed at IDE level and included at firmware level. The following libraries were included as shown below.

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include "DHT.h"
```

NodeMCU pins were defined for ease of understanding. The pin definitions as shown below were made as per the designed schematic diagram.

```
/******** Pin Definition ************/

//Relays for switching appliances
#define Relay1            D6
#define Relay2            D2
#define Relay3            D1
#define Relay4            D5

//DHT11 for reading temperature and
//humidity value
#define DHTPIN            D7

//buzzer to use it for alert
#define buzzer            D0
```

Then the user's Wi-Fi hotspot credentials were entered into the source code. Similarly, the username and a unique authorization key generated for the AdafruitIO MQTT account were added to the code. This key can be accessed by an authorized user only through the AdafruitIO dashboard created. So, this step was considered very important here, as, to establish the communication between the NodeMCU and the AdafruitIO Dashboard this key must be entered in the firmware as shown below. Next important step was to set the serial baud rate and configure the digital pins of the NodeMCU board as input and output and initializing the relays to a default state.

```
void MQTT_connect();
//Servo myservo;
void setup() {
  Serial.begin(115200);

  delay(10);

  pinMode(buzzer, OUTPUT);
  pinMode(Relay1, OUTPUT);
  pinMode(Relay2, OUTPUT);
  pinMode(Relay3, OUTPUT);
  pinMode(Relay4, OUTPUT);
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(A0, INPUT);
//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++//
digitalWrite(Relay1, HIGH);
digitalWrite(Relay2, HIGH);
digitalWrite(Relay3, HIGH);
digitalWrite(Relay4, HIGH);
//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++//
  Serial.println(F("Adafruit MQTT demo"));
```

10. EXPERIMENTAL RESULTS

In this section we had discussed about the hardware prototype developed for the experimentation and demonstration of the proposed work. The hardware prototype as could be seen below consisted of a 4-channel relay board with four LEDs interfaced to the GPIO port of the NodeMCU ESP8266 development board. The current sourcing and sinking capability of NodeMCU GPIO pins is limited and could not drive a relay directly due to its higher current demands. So, here for this purpose a current driver IC ULN2803 was interfaced in between the NodeMCU and Relay board. A buzzer was also interfaced to one of the pins of the NodeMCU board to show the connectivity status of NodeMCU with the Wi-Fi hotspot network. Small beeps after every 10 seconds indicated that the NodeMCU was searching around for the available Wi-Fi hotspot networks and a long beep of 3 seconds means it found one Wi-Fi hotspot network and got connected to it. In other words system got ready to talk to the network. A regulated DC power supply powered the system. As it could be interpreted from the figure that initially when the system got powered up and NodeMCU got connected to the local Wi-Fi hotspot, the relays were in default off state and hence all the electrical appliances (LED bulbs in this case) were initially off. The user had two options to get the relays turn on and turn off. One option was to control the switching of relays or say appliances by giving voice commands to a VPA i.e. Google Assistant in the smartphone. Another option was to control these relays from anywhere in the world by just logging into the AdafruitIO account and control the dedicated trigger button blocks on the dashboard. The relays or say appliances could be turned on and turned off one by one individually. There was another option to turn off the relays all together by using a single voice command in Google Assistant. The relays/ appliances could be controlled from AdafruitIO dashboard. The dashboard consists of four trigger-button blocks to monitor and control the switching of each relay individually.

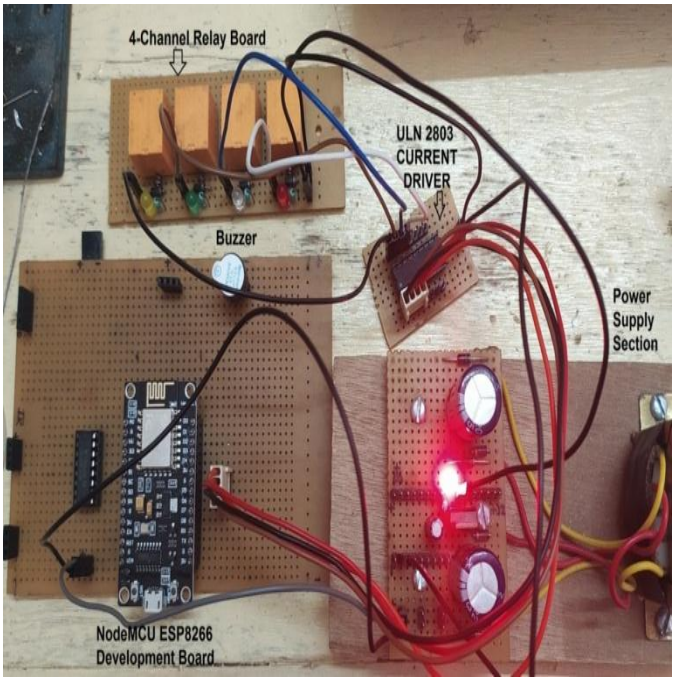

Fig -4: Experimental Setup

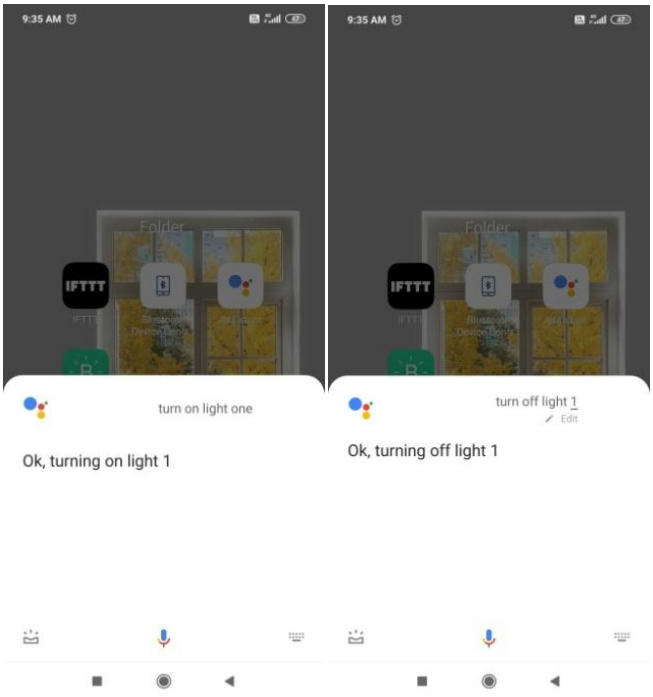

Fig-5: Turning Light-1 'On' & 'Off' using 'Google Assistant'

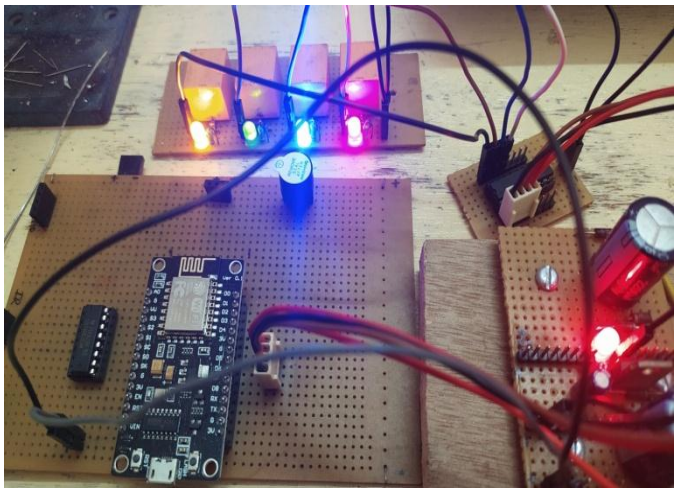

Fig.-6: Experimental Set-up during Testing Phase

Fig -7: System Status with all Appliances 'On'

Fig.-8: Default System Status on AdafruitIO Dashboard

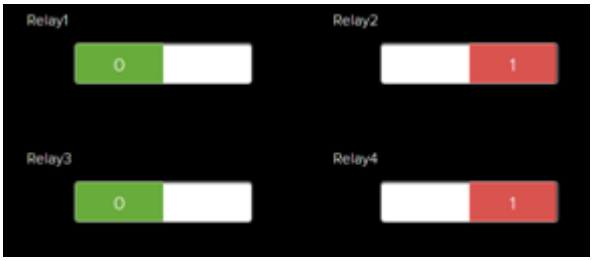

Fig.-9: Appliances in 'On' State & 'Off' State

11. CONCLUSION DRAWN

Finally an application has been implemented which included technologies like Artificial Intelligence based Voice Personal Assistant (VPA)

1. Transferring the data over to the Cloud via Google Assistant
2. Let the two apps connect via an IFTTT's Real-time API. Here, the created Applet connects Google Assistant to the Adafruit MQTT broker. It involved user-oriented triggers from the service that runs near-instantly
3. Proper message feeds were created by the user which declares every action that triggers based on the conditions fed by the user itself in the IFTTT applet
4. When a particular condition got satisfied, the Adafruit MQTT broker received feeds/ topics from the IFTTT client and sent it to the other client i.e. NodeMCU which was subscribed to that topic/ feed.
5. As the data received from the topic/ feed is '1', the Relay connected to the NodeMCU got turned ON and the LED bulb attached to it eventually turned ON.
6. At the same time the status of Relay block in the AdafruitIO dashboard got updated. Means the toggle button block got shifted to turn on position

Adafruit IO MQTT Dashboard

1. The NodeMCU here was a client and the Adafruit IO is a MQTT broker
2. All the relays were connected to the NodeMCU for monitoring and control the switching of appliances via the internet from anywhere in the world
3. Current status of all those relays were fed to different feeds/ topics on the same MQTT broker, in real-time only
4. User could monitor all the relay's status over the MQTT Dashboard on a laptop or a smartphone and also could send the control commands by just clicking on the GUI based Toggle Button on the Adafruit Dashboard

Hence, an IoT based system was implemented that utilizes another AI powered platform for controlling. So, Google Assistant controlled electrical appliances for home automation purpose was successfully implemented here in this work. The VPA is highly responsive in accepting commands and responding back with appropriate actions. This system provided user the liberty to use either a voice based command or a text command. The developed system was highly responsive and performance wise much more efficient than the conventional systems available.

12. FUTURE SCOPE

This work can be further extended to new levels as adoption rate of AI is increasing at a rapid pace and IoT has already captured the market. The combination of two would lead to the development and implementation of much more sophisticated systems. These systems would restrict human interventions as most of the tasks would be effectively and efficiently performed by these smart systems only. Google Assistant based controlling of different parameters and devices could be quite useful in the health sector, agriculture, defense sector, security, etc. One can integrate multiple sensors and devices with the help of IoT platforms and their controlling could be made easy using these AI powered VPAs like Google Assistant over the phone.

REFERENCES

[1] Haris Isyanto; AjibSetyo Arifin; Muhammad Suryanegara, "Design and Implementation of IoT-Based Smart Home Voice Commands for disabled people using Google Assistant", International Conference on Smart Technology and Applications (ICoSTA), 2020, Publisher: IEEE

[2] Tae-Kook Kim, "Short Research on Voice Control System Based on Artificial Intelligence Assistant", International Conference on Electronics, Information, and Communication (ICEIC), 2020, Publisher: IEEE

[3] Vanathi K. Lalitha; B. Mahalakshmi; S. Madhusudan; M. Srinivasaperumal; S. Srikanth; Sathish R. Kumar, "Smart Control Of Home Amenities Using Google Assistant And Clap Switch Circuit", 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, Publisher: IEEE

[4] David Sheppard; Nick Felker; John Schmalzel, "Development of Voice Commands in Digital Signage for Improved Indoor Navigation Using Google Assistant SDK", IEEE Sensors Applications Symposium (SAS), 2019, Publisher: IEEE

[5] Mokh. SholihulHadi; Maulana Ahmad As Shidiqi; Ilham Ari ElbaithZaeni; Muhammad Alfian Mizar; Mhd Irvan, "Voice-Based Monitoring and Control System of Electronic Appliance Using Dialog Flow API Via Google Assistant", International Conference on Electrical, Electronics and Information Engineering (ICEEIE), 2019, Volume-6, Publisher: IEEE

[6] İlkan Yıldırım; Erkan Bostancı; Mehmet Serdar Güzel, "Forensic Analysis with Anti-Forensic Case Studies on Amazon Alexa and GoogleAssistant Build-In Smart Home Speakers", 4th International Conference on Computer Science and Engineering (UBMK), 2019, Publisher: IEEE

[7] M. Sundarramurthi; A. M. Anjana Sundari; Anandi Giridharan, "A Method of Designing Home Automation Control System (HACS) Using Virtual Assistant And Mobile Application", International Conference on contemporary Computing and Informatics (IC3I), 2019, Publisher: IEEE

[8] Veton Këpuska; Gamal Bohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)", IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018, Publisher: IEEE

[9] Septimiu Mischie; Liliana Mâţiu-Iovan; Gabriel GăŠpăresc, "Implementation of Google Assistant on Raspberry Pi," International Symposium on Electronics and Telecommunications (ISETC), 2018, Publisher: IEEE

[10] Aleksandar Lazić; Milan Z. Bjelica; Dejan Nad; Branislav M. Todorović, "Google Assistant Integration in TV Application for Android OS", 26th Telecommunications Forum (TELFOR), 2018, Publisher: IEEE

[11] UrošVišekruna; Milan Savić, "Integration of Google Assistant in Android Application for Voice Control of Media Playback", 26th Telecommunications Forum (TELFOR), 2018, Publisher: IEEE